

In the Claims:

Please amend Claims 1, 13-15, 17, 59, 72-74, and 76; cancel Claims 16, 21-58, 75, and 80; and add new Claims 81-83, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A method for rendering a portal graphical user interface (GUI), comprising:

~~providing for the representation of a GUI desktop, a GUI look and feel, and a GUI book~~
as a set of controls, wherein the set of controls can be organized in a logical hierarchy, wherein each said control represents a corresponding graphical element in the portal GUI; wherein each said control has properties that can be read and set, and at least one said control can interact with another said control through an event notification mechanism, wherein each said control is implemented as one or more classes in an object-oriented programming paradigm, wherein each said control has one or more methods which can be overridden to provide specialization of the control;

traversing, using at least one processor, the representation, wherein the traversing comprises:

associating a first theme with a first control in the set of controls;

rendering the first control according to the first theme;

rendering any descendents of the first control according to the first theme unless the theme is overridden;

wherein overriding a second control, which is a descendent of the first control, ~~can override the theme~~ with a second theme such that the ~~descendent of the first second~~ control uses the second theme and any descendent of the ~~descendent of the first second~~ control uses the second theme unless the second theme is overridden ~~at the descendent of the descendent of the first control;~~ and

rendering the first control according to the first theme in parallel with rendering of the second control according to the second theme.

~~wherein one of the set of controls can communicate with another of the set of controls;~~

~~wherein controls represent corresponding graphical and functional elements in web applications; the controls have properties that can be read and set, and the controls can interact with each other through an event notification mechanism, the controls also have methods which provide services and which may be overridden to provide specialization of the control; controls~~

~~are implemented as one or more classes in an object-oriented programming paradigm to allow for new properties, events and/or specialized control methods to be provided by extending base control classes related to these features, at least some controls can serve as containers for other controls;~~

~~wherein at least two controls that are graphical elements in a web application that intercommunicate using the event notification mechanism; and~~

~~wherein the traversing step is done using at least one processor.~~

2. (Original) The method of claim 1 wherein:

the desktop is a view of a portal;

wherein the desktop can be represented by a desktop control; and

wherein the desktop control is hierarchically superior to the shell control and to the book control.

3. (Original) The method of claim 1 wherein:

the look and feel determines the appearance of the portal;

wherein the look and feel can be represented by a look and feel control; and

wherein the theme is a variation of the look and feel.

4. (Original) The method of claim 1 wherein:

the book can be used to navigate to at least one portal page; and

wherein the book is represented by a book control.

5. (Original) The method of claim 1 wherein:

one of the set of controls can respond to an event raised by another of the set of controls.

6. (Original) The method of claim 1 wherein:

a control can have an interchangeable persistence mechanism.

7. (Original) The method of claim 1 wherein:

a control can have an interchangeable rendering mechanism.

8. (Original) The method of claim 1, further comprising:

accepting a request.

9. (Original) The method of claim 8 wherein:
the request in a hypertext transfer protocol (HTTP) request.
10. (Original) The method of claim 8 wherein:
the request originates from a Web browser.
11. (Original) The method of claim 1, further comprising:
generating a response.
12. (Original) The method of claim 1 wherein:
a control can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button.
13. (Currently Amended) The method of claim 1 wherein:
associating the first theme with the first control can occur when the first control is rendered.
14. (Currently Amended) The method of claim 1 wherein:
the first control inherits the first theme from a parent control.
15. (Currently Amended) The method of claim 1 wherein:
the first theme specifies the appearance and/or functioning of [[an]] a control in the GUI.
16. (Canceled).
17. (Currently Amended) The method of claim 1 wherein:
the first theme can be specified in whole or in part by a properties file.
18. (Original) The method of claim 17 wherein:

the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5) Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX.

19. (Original) The method of claim 17 wherein:
the properties file can specify at least one image.

20. (Original) The method of claim 1 wherein:
the GUI is part of a portal on the World Wide Web.

21-58. (Canceled).

59. (Currently Amended) A machine readable storage medium having instructions stored thereon that when executed by a processor cause a system to:

provide ~~for the representation of GUI desktop, a GUI look and feel, and a GUI book as a~~ set of controls, wherein the set of controls are organized in a logical hierarchy, wherein each said control represents a corresponding graphical element in the portal GUI; wherein each said control has properties that can be read and set, and at least one said control can interact with another said control through an event notification mechanism, wherein each said control is implemented as one or more classes in an object-oriented programming paradigm, wherein each said control has one or more methods which can be overridden to provide specialization of the control;

traverse the representation, wherein the traversing comprises instructions to cause the system to:

associate theme with a first control in the set of controls;

render the first control according to the first theme;

render any descendents of the first control according to the first theme unless the theme is overridden;

~~wherein~~ override a second control, which is a descendent of the first control, ~~can override the theme~~ with a second theme such that the ~~descendent of the first~~ second control uses the second theme and any descendent of the ~~descendent of the first~~ second control uses the second theme unless the second theme is overridden ~~at the descendent of the descendent of the first control;~~ and

render the first control according to the first theme in parallel with rendering of the second control according to the second theme.

~~wherein one of the set of controls can communicate with another of the set of controls;
wherein controls represent corresponding graphical and functional elements in web applications; the controls have properties that can be read and set, and the controls can interact with each other through an event notification mechanism, the controls also have methods which provide services and which may be overridden to provide specialization of the control, controls are implemented as one or more classes in an object-oriented programming paradigm to allow for new properties, events and/or specialized control methods to be provided by extending base control classes related to these features, at least some controls can serve as containers for other controls.~~

60. (Previously Presented) The machine readable storage medium of claim 59, further comprising:

accepting a request.

61. (Previously Presented) The machine readable storage medium of claim 59 wherein:

the desktop is a view of a portal;

wherein the desktop can be represented by a desktop control; and

wherein the desktop control is hierarchically superior to the shell control and to the book control.

62. (Previously Presented) The machine readable storage medium of claim 59 wherein:

the look and feel determines the appearance of the portal;

wherein the look and feel can be represented by a look and feel control; and

wherein the theme is a variation of the look and feel.

63. (Previously Presented) The machine readable storage medium of claim 59 wherein:

the book can be used to navigate to at least one portal page; and

wherein the book is represented by a book control.

64. (Previously Presented) The machine readable storage medium of claim 59 wherein:

one of the set of controls can respond to an event raised by another of the set of controls.

65. (Previously Presented) The machine readable storage medium of claim 59 wherein:
a control can have an interchangeable persistence mechanism.
66. (Previously Presented) The machine readable storage medium of claim 59 wherein:
a control can have an interchangeable rendering mechanism.
67. (Previously Presented) The machine readable storage medium of claim 59, further comprising instructions that when executed cause the system to:
accept a request.
68. (Previously Presented) The machine readable storage medium of claim 67 wherein:
the request in a hypertext transfer protocol (HTTP) request.
69. (Previously Presented) The machine readable storage medium of claim 67 wherein:
the request originates from a Web browser.
70. (Previously Presented) The machine readable storage medium of claim 59, further comprising instructions that when executed cause the system to:
generate a response.
71. (Previously Presented) The machine readable storage medium of claim 59 wherein:
a control can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button.
72. (Currently Amended) The machine readable storage medium of claim 59 wherein:
associating the first theme with the first control can occur when the first control is rendered.
73. (Currently Amended) The machine readable storage medium of claim 59 wherein:
the first control inherits the first theme from a parent control.

74. (Currently Amended) The machine readable storage medium of claim 59 wherein:
the first theme specifies the appearance and/or functioning of [[an]] a control in the GUI.

75. (Canceled).

76. (Currently Amended) The machine readable storage medium of claim 59 wherein:
the first theme can be specified in whole or in part by a properties file.

77. (Previously Presented) The machine readable storage medium of claim 76 wherein:
the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5) Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX.

78. (Previously Presented) The machine readable storage medium of claim 76 wherein:
the properties file can specify at least one image.

79. (Previously Presented) The machine readable storage medium of claim 59 wherein:
the GUI is part of a portal on the World Wide Web.

80. (Canceled).

81. (New) The method of claim 1, further comprising:
using a first thread to render the first control, and a different thread to render another control in the set of controls.

82. (New) The method of claim 1, further comprising:
using a mainline render to obtain a render result for each control in the set of controls, if said control has been previously rendered by a separate render process.

83. (New) The method of claim 1, further comprising:
including content from an external site in each threads that is used to render one or more controls of the set of controls.